

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员 郑铭莉、陈佳欣、袁修竹**

**学 号 201530613801**

**201530611180**

**201530613535**

**邮 箱** [**727457905@qq.com**](mailto:727457905@qq.com)

**指导教师** 吴庆耀

**提交日期** **2017年 12 月 22 日**

## 实验题目: [基于AdaBoost算法的人脸检测](https://www.zybuluo.com/chenyaofo/note/960869)

## 2. 实验时间：2017年 12 月 16 日

## 3. 报告人: 郑铭莉、陈佳欣、袁修竹

## 4. 实验目的:

深入理解Adaboost的原理

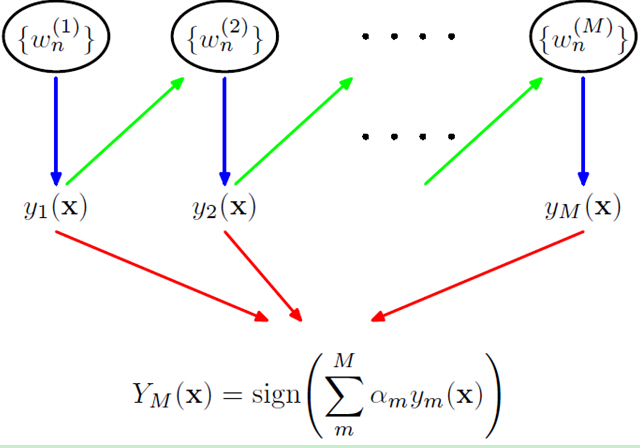
熟悉人脸检测的基本方法

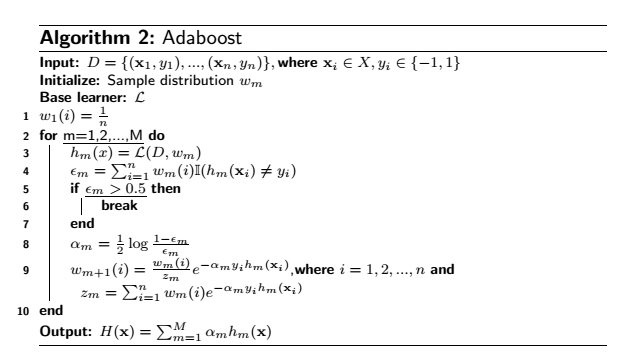
学会利用Adaboost解决人脸分类问题，将理论和实际工程接轨

体验机器学习的完整过程

## 5. 简述Adaboost原理：

AdaBoost是Boosting方法中最优代表性的提升算法。该方法通过在每轮降低分对样例的权重，增加分错样例的权重，使得分类器在迭代过程中逐步改进，最终将所有分类器线性组合得到最终分类器，Boost算法框架如下图所示：





## 6. 数据集以及数据分析：

本实验提供1000张图片，其中500张是含有人脸的RGB图片，储存在./datasets/original/face 内；另外500张是不含有人脸的RGB图，储存在./datasets/original/nonface 内。

数据集包含在示例仓库内，请自行下载并将其切分为训练集，验证集。

## 7. 实验步骤:

1.读取数据集数据。读取图片，将全部图片转成大小为24\*24的灰度图，数据集正负类样本的个数和比例不限，数据集标签形式不限。

2.处理数据集数据，提取NPD特征。使用feature.py中NPDFeature类的方法提取特征。（提示：因为预处理数据集的时间比较长，可以用pickle库中的dump()函数将预处理后的特征数据保存到缓存中，之后可以使用load()函数读取特征数据）

3.将数据集切分为训练集和验证集，本次实验不切分测试集。

4.根据ensemble.py中的预留的接口编写AdaboostClassifier所有函数。以下为AdaboostClassifier类中的fit()方法的思路：

4.1 初始化训练集的权值  ,每一个训练样本被赋予相同的权值。

4.2 训练一个基分类器，基分类器可以使用sklearn.tree库中DecisionTreeClassifier(注意训练的时候需要将权重  作为参数传入)。

4.3 计算基分类器在训练集上的分类误差率 ϵ 。

4.4 根据分类误差率 ϵ，计算参数 α 。

4.5 更新训练集的权值  。

4.6 重复以上4.2-4.6的步骤进行迭代，迭代次数为基分类器的个数。

5.用AdaboostClassifier中的方法在验证集上进行预测并计算精确率,并用sklearn.metrics库的classification\_report()函数将预测结果写入report.txt中。

6.整理实验结果并完成实验报告。

## 8. 代码内容:

训练过程：

for i in range(0, maxIteration):

print("Iteration",i)

clf\_tree.fit(img\_features\_train, img\_label\_train, sample\_weight=weights)

hypothesis.append (clf\_tree.predict(img\_features\_train) )

miss = [int(x) for x in ( hypothesis[i] != img\_label\_train )]

miss2 = [x if x==1 else -1 for x in miss]

err\_m = np.dot(weights,miss)

if(err\_m > 0.5):

break

print('error: ',err\_m)

alpha\_m.append( 0.5 \* np.log( (1 - err\_m) / float(err\_m)) )

weights = np.multiply(weights, np.exp([float(x) \* alpha\_m[i] for x in miss2]))

weights\_sum = weights.sum()

weights = weights / weights\_sum

prediction = prediction + alpha\_m[i] \* hypothesis[i]

print('train\_accuracy: ',get\_accuracy(np.sign(prediction),img\_label\_train))

validation\_pred = clf\_tree.predict(img\_features\_validation)

validation\_prediction = alpha\_m[i] \* validation\_pred

target\_names = ['class 0', 'class 1']

acc = get\_accuracy(np.sign(validation\_prediction), img\_label\_validation)

accuracy.append(acc)

print(classification\_report(img\_label\_validation, np.sign(validation\_prediction), target\_names=target\_names))

## 9. 实验结果和曲线图:

## 超参数选择（弱分类器个数、决策树深度等）：

分类器数：2

深度：3

## 预测结果（最佳结果）：

error: 0.00724637681159

train\_accuracy: 0.985714285714

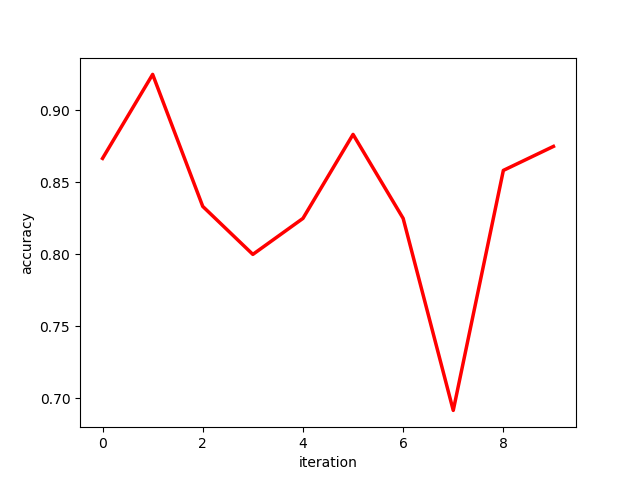
precision recall f1-score support

class 0 0.90 0.95 0.93 60

class 1 0.95 0.90 0.92 60

avg / total 0.93 0.93 0.92 120

## 精度曲线图：



## 10.实验结果分析：

训练集随着分类器数量增加，准确率逐渐提高，甚至达到1。但在验证集上的精确率在主要集中在0.7和0.8之间，当为两个分类器时达到精确率达到0.93。此外，深度达到4时出现计算溢出，因为error开始变为0。

## 11.实验总结：

对adaboost算法有了深刻的认识，它的分类效率很高，效果很好，但是对于决策树深度的认识还有疑惑，当深度在4之后都会出现计算溢出，原因还在查找。